Authentication of user accounts on OpenBSD using LDAP via RADIUS

What follows is a collection of notes made while configuring OpenBSD to authenticate user accounts against an LDAP directory via the RADIUS protocol.

The following are the steps I took to authenticate OpenBSD users against a RADIUS server running FreeRADIUS. FreeRADIUS has been installed and configured to support an LDAP backend. The LDAP backend is powered by OpenLDAP. Both daemons are running on a SparcStation 20 MP running Redhat Linux 6.2.

I have not tested the FreeRADIUS/OpenLDAP combination on OpenBSD (It's easier to work with Linux threads until OpenBSD's thread support matures). It should work, but YMMV.

1. Install and configure LDAP server.
2. Install, configure and test FreeRADIUS.

   Listed below are the places where I deviated from the default configuration of FreeRADIUS.

   - `/etc/radiusd.conf`

     ```
     lower_user = before
     lower_pass = no
     lower_time = before


     ldap {
             server = "localhost"
             basedn = "dc=webdaemons,dc=org"
             filter = "(ixAccount)(uid=%u))"
             start_tls = no
             access_group = "cn=eusers,ou=Groups,dc=webdaemons,dc=org"
             groupname_attribute = cn
             groupmembership_filter = "(|(upOfNames)\
             (member=%{Ldap-UserDn}))(upOfUniqueNames)\
             (uniquemember=%{Ldap-UserDn})))"
             timeout = 4
             timelimit = 3
             net_timeout = 1
             }

     # Authentication.
     #
     # This section lists which modules are available for authentication.
     # Note that it does NOT mean 'try each module in order'.  It means
     # that you have to have a module from the 'authorize' section add
     # a configuration attribute 'Auth-Type := FOO'.  That authentication type
     # is then used to pick the aproprіate module from the list below.
     #authenticate {
     #       pam
     #       unix
     #       ldap
     #       mschap
     #       eap
     #}
     authenticate {
             ldap
     }

     # Accounting. Log to detail file, and to the radwtmp file, and maintain
     # radutmp.
     accounting {
     #       acct_unique
     #       detail
     #       counter
     #       unix
     #       radutmp
     ```

```
#       sradutmp
}
```

- Populate `/etc/raddb/clients.conf` with your client's address and secret.

  ```
  client 192.168.100.100 {
          secret          = testing123
          shortname       = eris
  }
  ```

- `/etc/raddb/users`

  ```
  # First setup all accounts to be checked against the UNIX /etc/passwd.
  # (Unless a password was already given earlier in this file).
  #
  #DEFAULT        Auth-Type := System
  #       Fall-Through = 1

  DEFAULT Auth-Type := Ldap
          Fall-Through = 1

  #
  ```

3. Configure OpenBSD host with a user class using RADIUS as the authentication method.

   Example `login.conf` class for radius auth.

   ```
   eusers:\
           :requirehome@:\
           :auth=radius:\
           :radius-server=demeter.deadly.lab:\
           :radius-timeout=1:\
           :radius-retries=5:
   ```

4. Create local user on OpenBSD host with a login class of "eusers".

   Leave the password empty to prevent login from using the `/etc/passwd` entry. I like to test to make sure I *cannot* login.

   ```
    # useradd -m -d /home/euser -c "test radius user" -s /bin/ksh -u 10000 -L eusers euser
   ```

5. On your directory server, create an LDAP user, I use objectclass `posixAccount`.

   ```
   dn: uid=euser, ou=people, dc=webdaemons,dc=org
   sn: euser
   userPassword:: <password>
   loginShell: /bin/ksh
   l: Lincoln
   uidNumber: 10000
   gidNumber: 10000
   mail: euser@webdaemons.org
   objectClass: top
   objectClass: person
   objectClass: organizationalPerson
   objectClass: inetOrgPerson
   objectClass: posixAccount
   uid: euser
   gecos: test acct
   cn: euser
   homeDirectory: /home/euser
   description: test acct for radius auth
   st: Nebraska
   ```

6. Also create an LDAP access group, I use objectclass `posixGroup`, which also contains `groupOfUniqueNames`.

   ```
   dn: cn=eusers, ou=groups, dc=webdaemons,dc=org
   owner: cn=root,dc=webdaemons,dc=org
   ```

```
gidNumber: 10000
objectClass: top
objectClass: groupOfUniqueNames
objectClass: posixGroup
description: Authentication group for Remote user access
cn: eusers
memberUid: euser
uniqueMember: uid=euser,ou=People,dc=webdaemons,dc=org
```

If you *are* using access groups, make certain that the binddn in `radiusd.conf`, or in the case of anonymous bind operations, everyone has read access to your access groups `entry` and `memberUid` attributes. You should configure your LDAP acl's based on your security policy.

7. Populate `/etc/raddb/servers` on the OpenBSD host with the hostname & shared secret of the RADIUS server.

```
 # mkdir -m 755 /etc/raddb

 # echo "demeter.deadly.lab testing123" > /etc/raddb/servers

 # chmod 400 /etc/raddb/servers
```

8. Test.

```
[jamesp@artemis jamesp]$ ssh -l euser eris
euser@eris's password:
Last login: Sun Apr 21 17:54:59 2002 from artemis.deadly.org
OpenBSD 3.0-stable (GENERIC) #0: Sun Mar 17 00:35:47 GMT 2002

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ id
uid=10000(euser) gid=10(users) groups=10(users)
$
```

This is a preliminary document. Additions are welcome from the user community. Please send questions, comments, and additions to jamesp@webdaemons.org

$Id: openbsd_ldap.shtml,v 1.6 2003/04/16 04:12:25 jamesp Exp $

All content presented here is for informational purposes, no guarantee of suitibility for any purpose, either expressed or implied is given.